HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

PATENT APPLICATION

ATTORNEY DOCKET NO. 100203448-2

IN THE
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s):	Bradley G. Culter	Confirmation No.:	5828
Application No.:	10/685,287	Examiner:	M. J. Yigdall
Filing Date:	October 14, 2003	Group Art Unit:	2192

Title: ALLOWING FIRMWARE TO BORROW A PROCESSOR

Mail Stop Appeal Brief - Patents
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF REPLY BRIEF

Transmitted herewith is the Reply Brief with respect to the Examiner's Answer mailed on 05/29/2008.

This Reply Brief is being filed pursuant to 37 CFR 1.193(b) within two months of the date of the Examiner's Answer.

(Note: Extensions of time are not allowed under 37 CFR 1.136(a))

(Note: Failure to file a Reply Brief will result in dismissal of the Appeal as to the claims made subject to an expressly stated new ground rejection.)

No fee is required for filing of this Reply Brief.

If any fees are required please charge Deposit Account 08-2025.

I hereby certify that this document is being
transmitted to the Patent and Trademark Office
via eletronic filing.

Date of Transmission: 7/9/2008

Donna Forbit

Respectfully submitted,
Bradley G. Culter

By:

Jody C. Bishop
Attorney/Agent for Applicant(s)

Reg No. : 44,034
Date : July 9, 2008
Telephone : 214-855-8007

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

Docket No.: 100203448-2
(PATENT)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Bradley G. Culter

Application No.: 10/685,287

Confirmation No.: 5828

Filed: October 14, 2003

Art Unit: 2192

For: ALLOWING FIRMWARE TO BORROW A PROCESSOR

Examiner: M. J. Yigdall

REPLY BRIEF

MS Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

As required under § 41.41(a)(1), this Reply Brief is filed within two months of the Examiner's Answer dated May 29, 2008, and is in furtherance of the Appeal Brief filed on March 14, 2008.

No fee is required for this REPLY BRIEF.

This brief contains items under the following headings pursuant to M.P.E.P. § 1208:

I. Status of Claims
II. Ground of Rejection to be Reviewed on Appeal
III. Argument
IV. Conclusion

I. STATUS OF CLAIMS

The status of claims remains as identified in the Appeal Brief submitted March 14, 2008.

II. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The grounds of rejection to be reviewed remain as identified in the Appeal Brief submitted March 14, 2008.

III. ARGUMENT

Appellant respectfully traverses the outstanding rejections of the pending claims and requests that the Board overturn those rejections in light of the remarks contained in Appellant's Appeal Brief of March 14, 2008 (*Appeal Brief*) and in this Reply Brief. The claims separately argued in the *Appeal Brief* do not stand or fall together. *See* 37 C.F.R. § 41.31(c)(1)(vii). Appellant hereby reasserts the arguments previously presented in the *Appeal Brief*. For the sake of brevity, however, Appellant does not restate those same arguments herein, but instead submits the following supplemental remarks in response to the arguments and remarks contained in the Examiner's Answer.

A. First Ground of Rejection

Claims 1-6, 8, 10-12, 14-20, 23-26, and 28-34 are rejected under 35 U.S.C. § 102(e) as being anticipated by O'Shea (U.S. Application 2004/0128568). Appellant maintains that this ground of rejection should be overturned for the reasons discussed in the *Appeal Brief* and further below.

1. Claims 1-6, 8, 10-12, and 14

Independent claim 1 recites:

> A method for changing control of a processor that is in an active state under the control of an operating system to a borrowed state wherein the processor is under control of firmware, comprising:
> sending a request for a change in control to the operating system;
> deciding, by the operating system, whether to grant the request;
> placing the processor in a transitional state that is different from the active state, if the request is granted; and
> sending, by the operating system, an interrupt signal to move the processor from the transitional state into the borrowed state.

In the *Appeal Brief*, Appellant argued that *O'Shea* does not teach the recited "deciding" and "sending" operations that are recited in claim 1 as being performed by the operating system ("OS"), *see* pages 6-8 of the *Appeal Brief.* Appellant argued that *O'Shea* teaches that, rather than the OS deciding whether to grant a request to move a processor to a borrowed state, "usurping code" is employed to usurp (or "steal") control of the processor from the OS. Thus, rather than the OS making the decision to grant its processor to be "borrowed," *O'Shea* teaches that "usurping code" can steal control of the processor away from the OS.

In response to the arguments presented in the *Appeal Brief*, the Examiner's Answer argues that power management code included in *O'Shea's* OS is leveraged for usurping control of the processor, *see* pages 17-21 of the Examiner's Answer. The Examiner's Answer argues that it is thus the power management code included in the OS that decides whether to usurp control of the processor to firmware. Also, the Examiner's Answer argues that there is no difference between *O'Shea's* usurping of the processor and claim 1's recitation of the OS sending a signal to move the processor into a "borrowed state" because in *O'Shea* control of the processor is eventually returned to the OS, *see* pages 17-21 of the Examiner's Answer.

While much of the Examiner's characterization of *O'Shea* initially appears to be accurate, the Examiner's characterization of *O'Shea* as disclosing that its OS decides to move the processor to a "borrowed state" is inaccurate, as discussed below. *O'Shea* discloses power management code, which may be integrated in the OS, *see e.g.*, paragraph 0022 of *O'Shea*. The power management code can initiate entry into a lower power state. When entering a lower power state, the OS saves certain configuration and state information and then temporarily halts its execution of code (for power conservation), *see* paragraph 0024 of *O'Shea*. *O'Shea* leverages that feature of the power management code to, once the OS enters a lower power state, usurp control of the system away from the OS. For instance, paragraph 0023 of *O'Shea* explains:

> At some point during normal use of computer system 100, either action by a user of computer system 100 or other stimulus causes executable code for software to usurp control of computer system 100 from the operating system to also be loaded and executed to the extent necessary to enable control of computer system 100 to be usurped from operating system 100 in a manner that uses the mechanism that is normally used for placing computer system 100 in a lower power state. Essentially, this code for usurping control defines memory locations where coordinating information may be placed and/or readies pieces of executable code that aids in taking over control of computer system 100 from the operating system and giving control over to executable code within firmware 150 at a time when code for power management would normally cause the temporary halting of execution of the code for the operating system as part of causing computer system 100 to enter a lower power state.

Once control is usurped from the OS in *O'Shea*, code unrelated to power conservation can be executed by the firmware. For instance, paragraph 0025 of *O'Shea* explains:

> In some embodiments, it is at the point at which control of computer system 100 has been relinquished to executable code within firmware storage 150 for purposes of preparation to enter a lower power state that executable code within firmware 150 actually uses the relinquished control of computer system 100 to usurp of 'steal away' computer system 100 from the operating system to perform a task unrelated to preparations for entering a lower power state, instead.

Control can then be returned to the OS just as it is normally returned when exiting the lower power state (e.g., using the saved configuration and state information). For instance, paragraph 0025 of *O'Shea* further explains:

> Upon completion of execution of such code to perform one or more of such tasks, control of computer system 100 is returned to the operating system

> just as if code within firmware storage 150 for making preparations for computer system 100 to enter a lower power state had been fully executed.

Thus, in *O'Shea* usurping code (external to the OS) can request to usurp control of the system. When the OS is entering a lower power state, a determination is made whether such a usurping request is present. If not, the OS enters its lower power state in a normal manner, *see* blocks 320, 330, 340, 360, 370, and 380 of FIG. 3 and paragraphs 0041-0044 of *O'Shea*. If a usurping request is present, then, after the OS prepares to enter its lower power state, control of the system is usurped by firmware to execute other code. *Id.*

Accordingly, in *O'Shea* the OS (e.g., the power management code) does NOT decide whether to grant a request to change control of a processor, but instead the OS merely decides to enter a lower power state. The OS may relinquish control to firmware for the purpose of performing tasks for entering the lower power state, but the OS does not decide to change control of a processor to a borrowed state. Even if the relinquishing of control from the OS to the firmware for the purpose of performing tasks for entering a lower power state is considered as "placing the processor in a transitional state that is different from the active state" (as recited by claim 1), the OS of *O'Shea* does NOT send an interrupt signal to move the processor from the transitional state into the borrowed state. Instead, in *O'Shea* the external usurping code (of the firmware) decides whether to change control of the processor by issuing a signal to usurp or "steal" control away from the OS when the OS enters its lower power state.

In this manner, *O'Shea* teaches a system that intelligently chooses a time during which to steal control of the processor (i.e., when the OS enters a lower power state) so that the processor can be returned to the OS with the configuration and state information preserved by the OS, rather than involving the OS in the borrowing decision-making as recited by claim 1. *O'Shea* employs this technique for the express purpose of stealing away control of the processor without involving the OS in the decision-making process. In other words, rather than the OS deciding whether to grant a request to change control of the processor, in *O'Shea* the OS simply decides to enter a lower power state, and the usurping code takes advantage of the fact that the OS saves its configuration and state information when entering the lower power state in choosing this time to "steal away" control of the processor from the OS, *see e.g.*, paragraphs 0014 and 0023-0025 of *O'Shea*.

Additionally, such usurping of the processor control in *O'Shea* is different than a "borrowed" state. Even though the processor control is ultimately returned to the OS in *O'Shea*, the OS does not decide that the processor control could be borrowed in the first instance. Rather, from a review of *O'Shea*, one finds that *O'Shea* consistently and repeatedly emphasizes that the usurping code "steals away" control of the processor, and then later returns control to the OS. *O'Shea* makes clear repeatedly that the OS merely enters a lower power state, and when entering such lower power state the usurping code "steals away" control of the processor. Thus, the OS does NOT send a signal to move the processor into a borrowed state (as recited by claim 1), but instead merely enters a lower power state and has its processor control stolen.

The operation of *O'Shea* is analogous to a shoplifter stealing from a retailer while the retailer is closed (or "sleeping", such as when the OS is in a lower power state). While the shoplifter may later argue that he/she was only "borrowing" the stolen item because he/she was intending to return the item when the retailer re-opened, in the retailer's view the item was not borrowed, but was stolen without permission - merely because the retailer was closed and not using the item does not grant permission for unauthorized shoplifters to "steal away" the item. This is consistent with *O'Shea's* repeated characterization of its operation as usurping or stealing away control of the processor from the OS when the OS enters a lower power state. Indeed, as discussed above, *O'Shea* intentionally chooses a time during which to steal control of the processor (i.e., when the OS enters a lower power state) so that the processor can be returned to the OS with the configuration and state information preserved by the OS, This time selection is required in *O'Shea* because the OS is NOT involved in the borrowing decision-making as recited by claim 1. This is analogous to a shoplifter intentionally choosing a time during which a retailer is closed in which to steal an item from the retailer. Even though the shoplifter may intend to return the item prior to the retailer re-opening for business (as *O'Shea* intends to return control of the processor prior to the OS resuming operation from its lower power state), this action is not fairly characterized as "borrowing" of the item by the shoplifter because the retailer was not involved and did not grant permission for the item to be "borrowed". Instead, just as *O'Shea* properly characterizes this action, it is stealing rather than borrowing.

Thus, claim 1 is not anticipated by *O'Shea* because *O'Shea* fails to teach "deciding, by the operating system, whether to grant the request" for a change in control, and because *O'Shea* fails to teach "sending, by the operating system, an interrupt signal to move the processor from the transitional state into the borrowed state." Indeed, as discussed above, the OS of *O'Shea* does not send any such interrupt signal to move the processor from a transitional state into a borrowed state, but instead merely enters a lower power state at which point external usurping code steals away control of the processor. Therefore, Appellant maintains that the present rejection is improper and should be overturned.

Claims 2-6, 8, 10-12, and 14 depend, either directly or indirectly, from claim 1 and thus, inherits the limitations of claim 1. As a result, claims 2-6, 8, 10-12, and 14 are allowable for at least the reasons set forth above with respect to independent claim 1.

2. Claims 15-20, 23-26, and 28-30

Claim 15 defines a computer system wherein the operating system decides whether to place the processor in the borrowed state and sends an interrupt signal to move the processor into the borrowed state. *O'Shea* does not disclose at least these limitations. As discussed in detail above for claim 1, *O'Shea* fails to teach the OS deciding weather to move the processor into a borrowed state or sending an interrupt signal to move the processor into the borrowed state. Therefore, the Appellant respectfully asserts that for the above reasons claim 15 is patentable over the 35 U.S.C. § 102 rejection of record, and thus the rejection should be overturned.

Claims 16-20, 23-26, and 28-30 depend, either directly or indirectly, from claim 15 and thus, inherits the limitations of claim 15. As a result, claims 16-20, 23-26, and 28-30 are allowable for at least the reasons set forth above with respect to independent claim 15.

Appellant further asserts that there are many limitations contained in the dependent claims that are patentable in their own right. For example, claim 28 recites "wherein the operating system decides without involvement of the firmware whether to place the processor in the borrowed state." As discussed above, there is no teaching in *O'Shea* of the operating system having any part in the decision making. Instead, it is the usurping code (included in the firmware) of *O'Shea* that is making the decision. It is clear that there is no teaching that the operating system is making the entire decision without involvement of the firmware in *O'Shea*. Thus, claim 28 does not stand or fall with claim 15, but is further patentable over *O'Shea* for this additional reason.

3. Claims 31-32

Claim 31 defines a computer system that has a processor for executing code, and includes means, operative by the operating system, for deciding whether to grant a request to change control of a processor from an active state to a borrowed state. As discussed above, *O'Shea* fails to teach any such means in its operating system for deciding whether to grant a request to change control of a processor from an active state to a borrowed state. Instead, the OS in *O'Shea* merely decides to enter a lower power state, and then when in the lower power state external usurping code can decide to steal away control of the processor. At no point does the OS of *O'Shea* change control of a processor to a borrowed state, as recited by claim 31. Therefore, the Appellant respectfully asserts that for the above reasons claim 31 is patentable over the 35 U.S.C. § 102 rejection of record, and thus the rejection should be overturned.

Claim 32 depends from claim 31 and thus, inherits the limitations of claim 31. As a result, claim 32 is allowable for at least the reasons set forth above with respect to independent claim 31.

4. Claims 33-34

Claim 33 defines a computer readable medium having computer program logic recorded thereon for changing control of a processor that is in an active state under the control of an operating system to a borrowed state wherein the processor is under control of firmware. The medium includes logic for deciding, by the operating system, whether to grant the request. As discussed above, *O'Shea* fails to teach any such logic for deciding whether to grant such a request. Instead, the OS in *O'Shea* merely decides to enter a lower power state, and then when in the lower power state external usurping code can decide to steal away control of the processor. At no point does the OS of *O'Shea* decide to change control of a processor to a borrowed state, as recited by claim 33. Therefore, the Appellant respectfully asserts that for the above reasons claim 33 is patentable over the 35 U.S.C. § 102 rejection of record, and thus the rejection should be overturned..

Claim 34 depends from claim 33 and thus, inherits the limitations of claim 33. As a result, claim 34 is allowable for at least the reasons set forth above with respect to independent claim 33.

B. Second Ground of Rejection

Appellant respectfully reiterates the arguments presented in the *Appeal Brief* for the second ground of rejection.

C. Third Ground of Rejection

Appellant respectfully reiterates the arguments presented in the *Appeal Brief* for the third ground of rejection.

IV. CONCLUSION

Appellant respectfully requests that the Board overturn the rejections of pending claims 1-34 for the reasons stated in the *Appeal Brief* and herein.

Dated: July 9, 2008

Respectfully submitted,

By____________________
Jody C. Bishop
Registration No.: 44,034
Attorney for Applicant
(214) 855-8007

I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being transmitted via the Office electronic filing system in accordance with § 1.6(a)(4).

Dated: July 9, 2008

Signed: Donna Forbit
Donna Forbit